MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 3, 2018/2019

ECP2216 – MICROCONTROLLER AND MICROPROCESSOR SYSTEMS

(TE / RE)

27 MAY 2019
2.30 p.m. - 4.30 p.m.
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 10 pages with 5 Questions only.

2. Attempt **ALL FIVE COMPULSORY** questions. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please write all your answers in the Answer Booklet provided.

Question 1

a) Using a block diagram, explain what a von Neuman architecture is.

[5 marks]

b) i) Find the range of an 8-bit two's complement number.

[2 marks]

ii) Convert 88_{10} and -14_{10} into 8-bit two's complement number.

[3 marks]

iii) Add the two 8-bit two's complement numbers found in part (b)(ii) and show the result as a decimal number.

[2 marks]

c) Compare **THREE** features of Complex Instruction Set Computer (CISC) and Reduced Instruction Set Computer (RISC). Give an example of each RISC and CISC computer.

[5 marks]

d) Native 80386 processors can operate either in Real-addressed mode or Protected-addressed mode. Give **THREE** advantages of protected-addressed mode over real-addressed mode.

[3 marks]

Continued...

Question 2

a) Evaluate the bit addresses that are set to '1', after the execution of the following instruction.

MOV 28H, #35H

[2 marks]

b) Figure out the contents of Accumulator (ACC), B register, R0 register, and Program Status Word (PSW) after the execution of each instructions, assuming the initial value for ACC = 88H, B = AAH, R0 = AEH, and PSW = 00H.

```
MOV   R0, #0B3H
MOV   A, #13H
ADD   A, R0
MOV   B, #00000011B
MUL   AB
```

Instruction	ACC	B	R0	PSW
Initial	88H	AAH	AEH	00H
MOV R0, #0B3H				
MOV A, #13H				
ADD A, R0				
MOV B, #00000011B				
MUL AB				

[7 marks]

c) Design an 8051-microcontroller based system that can address 16 Kbytes of data memory and specify the address range of each memory block used. Given that the available memory ICs are 4 Kbytes ROM and 4 Kbytes RAM.

[11 marks]

Continued...

Question 3

a) Determine the addressing modes for the source and the destination operand for the instructions below.

i) ORL R0, #0F0H
ii) MOVX @R0, A
iii) MOVC A, @A+PC

[3 marks]

b) Given the sequence of MCS-51 assembly language program, find the machine codes, execution time and the code size.

```
ORG   1000H
MOV   A, P0
ANL   A, #0FH
MOV   R0, A
NOP
MOV   A, P2
ANL   A, #0FH
ORL   A, R0
MOV   P0, A
```

[8 marks]

c) i) Write an MCS-51 assembly language subroutine that gives roughly 2ms delay. Name the subroutine as "DELAY_2MS". Assume that the microcontroller is using 12 MHz clock.

[5 marks]

ii) Using the subroutine "DELAY_2MS" in Q3(c)(i), write an MCS-51 assembly language program that generates 0.1 second delay. Assume that microcontroller is using 12 MHz clock.

[4 marks]

Continued...

Question 4

a) Using a timer in an 8051, write an MCS-51 assembly language program to generate a Pulse Width Modulated (PWM) waveform as shown in Fig. 4a and output the waveform in P1.0. Assume that the microcontroller is using 12 MHz crystal clock.

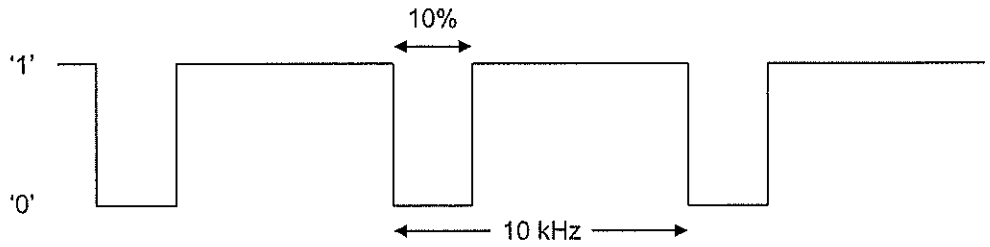


Fig. 4a

[12 marks]

b) Write an MCS-51 assembly language program to configure an 8051 serial port with 9600 baud rate, 8-bit data, 1 stop bit, and no parity. Assume that the microcontroller is running with 11.0592 MHz clock.

[8 marks]

Continued...

Question 5

a) Given a T-junction crossroad as depicted in Fig. 5a.

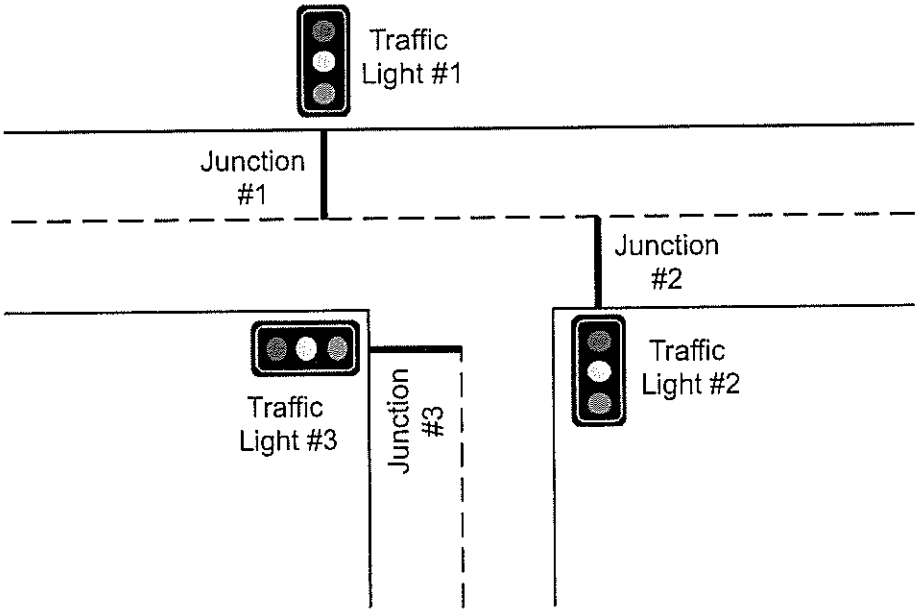


Fig. 5a

Write an MCS-51 assembly language program for the traffic light controller with the following ON/OFF duration.

Traffic Light	Yellow Light Duration (s)	Green Light Duration (s)
#1	5	75
#2	5	75
#3	5	25

Vehicles in Junction #1 should go first, followed by vehicles in Junction #2, and then vehicles in Junction #3. Assume that the traffic light control logic is active low and the I/O connection is as following.

Traffic Light	Red Light	Yellow Light	Green Light
#1	P2.0	P2.1	P2.2
#2	P2.3	P2.4	P2.5
#3	P0.0	P0.1	P0.2

[20 marks]

End of Page

APPENDIX

Special Function Register Formats

Interrupt Enable (IE)

Bit Addr.	AFH	-	-	ACH	ABH	AAH	A9H	A8H
Name	**EA**	-	-	**ES**	**ET1**	**EX1**	**ET0**	**EX0**

BIT	SYMBOL	FUNCTION (Enable=1, Disable=0)
IE.7	EA	Global enable/disable. EA = 1, each individual source is enabled/disabled by setting/clearing its enable bit. EA = 0, disable all interrupts.
IE.6	-	Undefined
IE.5	-	Not implemented in 8051. ET2 for 8052.
IE.4	ES	Serial port interrupt enable bit.
IE.3	ET1	Timer 1 interrupt enable bit.
IE.2	EX1	External interrupt enable bit.
IE.1	ET0	Timer 0 interrupt enable bit.
IE.0	EX0	External interrupt enable bit.

Interrupt Priority (IP)

Bit Addr.	-	-	-	BCH	BBH	BAH	B9H	B8H
Name	-	-	-	**PS**	**PT1**	**PX1**	**PT0**	**PX0**

BIT	SYMBOL	FUNCTION (Enable=1, Disable=0)
IP.7	-	Undefined
IP.6	-	Undefined
IP.5	-	Not implemented in 8051. PT2 for 8052.
IP.4	PS	Serial port interrupt priority bit.
IP.3	PT1	Timer 1 interrupt priority bit.
IP.2	PX1	External interrupt priority bit.
IP.1	PT0	Timer 0 interrupt priority bit.
IP.0	PX0	External interrupt priority bit.

Interrupt Vectors

Interrupt Source	Flag	Vector Address
System Reset	RST	0000H
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RI & TI	0023H
Timer 2 (8052)	TF2 or EXF2	002BH

Program Status Word (PSW)

Bit Addr.	D7H	D6H	D5H	D4H	D3H	D2H	-	D0H
Name	**CY**	**AC**	**F0**	**RS1**	**RS0**	**0V**	**-**	**P**

Serial Control (SCON)

Bit Addr.	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
Name	**SM0**	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

BIT	**SYMBOL**	**FUNCTION**
SCON.7	SM0	Serial port mode bit 0 (see Table A.1).
SCON.6	SM1	Serial port mode bit 1 (see Table A.1).
SCON.5	SM2	Serial port mode bit 2; enables multiprocessor communications in modes 2 and 3; RI will not be activated if received 9th bit is 0. In mode 1, if SM2 = 1, then RI will be activated only if a valid stop bit was received. In mode 0, SM2 should be 0.
SCON.4	REN	Receiver enable; must be set to receive characters.
SCON.3	TB8	Transmit bit 8; 9th bit transmitted in modes 2 and 3; set/cleared by software.
SCON.2	RB8	Receive bit 8; 9th bit received.
SCON.1	TI	Transmit interrupt flag; set at end of character transmission; cleared by software.
SCON.0	RI	Receive interrupt flag; set at end of character reception; cleared by software.

Table A.1 The 8051 Serial Port Mode Selection

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift register	Fixed
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fixed
1	1	3	9-bit UART	Variable

Timer Control (TCON)

Bit Addr.	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

BIT	SYMBOL	FUNCTION
TCON.7	TF1	Timer-1 overflow flag. Set by hardware on overflow. Cleared by hardware when processor vectors to interrupt routine. Must be cleared by software when not involve interrupt
TCON.6	TR1	Timer-1 run control bit. Set/cleared by software to turn timer/counter on/off.
TCON.5	TF0	Timer-0 overflow flag. Do the same function as TF1 but for Timer-0.
TCON.4	TR0	Timer-0 run control bit. Do the same function as TR1 but for Timer-0.
TCON.3	IE1	External interrupt-1 edge flag. Set by hardware when interrupt-1 falling edge is detected. Cleared by hardware when interrupt is processed.
TCON.2	IT1	Interrupt-1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TCON.1	IE0	External interrupt-1 edge flag. Do the same function as IE1 but for external interrupt-0.
TCON.0	IT0	Interrupt-0 Type control bit. Do the same function as IT1 but for external interrupt-0.

Timer Mode (TMOD)

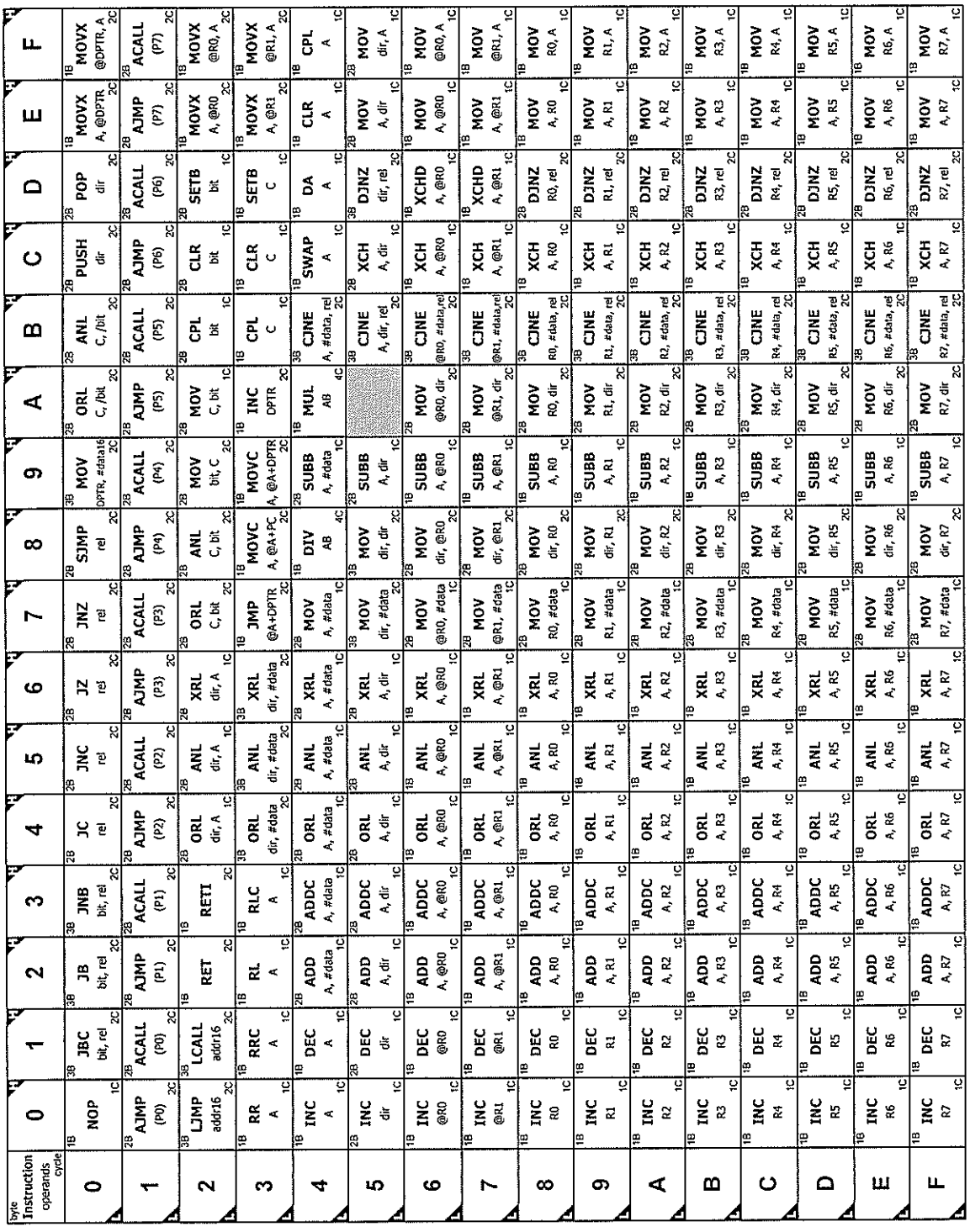
Bit	7	6	5	4	3	2	1	0
Name	GATE	C/T	M1	M0	GATE	C/T	M1	M0

BIT	SYMBOL	FUNCTION
TMOD.7	GATE1	When this bit is set the timer will only run when INT1 (P3.3) is high (hardware control). When this bit is cleared the timer will run regardless of the state of INT1 (software control).
TMOD.6	C/T1	Timer / Counter select bit. C / $\overline{T}$ = 0 → Timer operation. C / $\overline{T}$ = 1 → Counter operation.
TMOD.5	M1	Mode selection bits (see Table A.2). [for timer 1]
TMOD.4	M0	Mode selection bits (see Table A.2). [for timer 1]
TMOD.3	GATE0	Exactly the same function as GATE1 but for Timer0
TMOD.2	C/T0	Exactly the same function as C/T1 but for Timer0
TMOD.1	M1	Mode selection bits (see Table A.2). [for timer 0]
TMOD.0	M0	Mode selection bits (see Table A.2). [for timer 0]

Table A.2 Timer Mode Selection

M1	M0	Timer Mode	Description of Mode
0	0	0	13-bit Timer
0	1	1	16-bit Timer
1	0	2	8-bit auto-reload
1	1	3	Split timer mode

MCS-51 Opcode Map


byte
Instruction operands
cycle
0 1 2 3 4 5 6 7 8 9 A B C D E F
0 NOP JBC JB JNB JC JNC JZ JNZ SJMP MOV ORL ANL PUSH POP MOVX MOVX
1 AJMP ACALL AJMP ACALL AJMP ACALL AJMP ACALL AJMP ACALL AJMP ACALL AJMP ACALL AJMP ACALL
2 LJMP LCALL RET RETI ORL ANL XRL JMP MOVC ANL MOV MOV CPL SETB MOVX MOVX
3 RR RRC RL RLC ORL ANL XRL MOVC DIV SUBB MUL CJNE SWAP DJNZ XCH XCHD MOVX
4 INC DEC ADD ADDC ORL ANL XRL MOV SUBB INC CJNE CLR DJNZ MOV
5 INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH DJNZ MOV
6 INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH XCHD MOV
7 INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH XCHD MOV
8 INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH DJNZ MOV
9 INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH DJNZ MOV
A INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH DJNZ MOV
B INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH DJNZ MOV
C INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH DJNZ MOV
D INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH DJNZ MOV
E INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH DJNZ MOV
F INC DEC ADD ADDC ORL ANL XRL MOV SUBB MOV CJNE XCH DJNZ MOV